

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR UNITED STATES PATENT
FOR

PRIORITY FORWARDING IN A COMMUNICATION SYSTEM

Inventors:

Bradley Cain

Larry DiBurro

Bill Miller

Mike Berger

Bob Lee

Attorney Docket: 2204/179 (BA407)

Attorneys:

BROMBERG & SUNSTEIN LLP
125 Summer Street
Boston, MA 02110
(617) 443-9292

PRIORITY FORWARDING IN A COMMUNICATION SYSTEM

5

FIELD OF THE INVENTION

The present invention relates generally to communication systems, and more particularly to priority forwarding of protocol messages among layers in a protocol stack.

10

BACKGROUND OF THE INVENTION

In today's information age, the various network devices in a communication network utilize numerous protocols for exchanging information. These protocols are typically implemented in layers, which are often referred to collectively as a "protocol stack." Each protocol layer in the protocol stack provides services to the protocol layer above it in the protocol stack and consumes services from the protocol layer below it in the protocol stack.

Many protocol stacks use a message passing architecture for passing protocol messages between protocol layers. In particular, queues are typically used to transfer protocol messages between a particular protocol layer and its adjacent protocol layers. When a lower layer protocol needs to forward a protocol message to an upper layer protocol, the lower layer protocol places the protocol message on a queue corresponding to the upper layer protocol.

One advantage of such a message passing architecture is that the protocol message order is maintained. Since a queue is a first-in-first-out (FIFO) mechanism, protocol messages are removed from the queue in the same order that the protocol messages are placed into the queue. The lower layer protocol places protocol messages into the queue in the order in which the protocol messages are received. Therefore, the upper layer protocol removes the protocol messages from the queue in the order in which the protocol messages are received.

One disadvantage of such a message passing architecture is that there is no consideration for protocol message priority. Almost invariably, certain types of protocol

messages are more important than others, and are therefore preferably processed ahead of less important protocol messages. Unfortunately, when a queue is used for message passing, the lower layer protocol places high priority protocol messages and low priority protocol messages on the queue in the order in which they are received, and therefore the high priority protocol messages continue to be processed in the order in which they are received rather than being processed ahead of earlier low priority messages.

One way to provide priority forwarding for protocol messages is to utilize two (2) queues, namely a high priority message queue and a low priority message queue. The lower layer protocol places high priority protocol messages on the high priority message queue, and places low priority protocol messages on the low priority message queue. The upper layer protocol processes all protocol messages in the high priority message queue before processing any protocol messages in the low priority queue. In this way, the high priority protocol messages are processed ahead of earlier low priority messages.

In order to support such a priority forwarding mechanism, the lower layer protocol must be able to determine a priority for each protocol message that it forwards to the upper layer protocol. Typically, the lower layer protocol logic is modified to interpret certain protocol elements of the upper layer protocol in order to determine a priority for each protocol message. This adds complexity to the lower layer protocol, and makes the upper layer protocol dependent on the specific lower layer protocol for receiving priority forwarding services.

Thus, there is a need for an improved priority forwarding mechanism.

SUMMARY OF THE INVENTION

In accordance with one aspect of the invention, an upper layer protocol that requires priority forwarding includes a priority function that is invoked by a lower layer protocol in order to obtain the priority for a protocol message. The upper layer protocol provides the priority function to the lower layer protocol when the upper layer protocol is installed in the protocol stack. The lower layer protocol invokes the priority function before passing a protocol message to the upper layer protocol in order to determine a

5

priority for the protocol message. The lower layer protocol places the protocol message in the appropriate queue based upon the priority indicator obtained from the upper layer protocol's priority function. In this way, the lower layer protocol is not required to interpret any protocol elements of the upper layer protocol in order to determine a priority for each protocol message, and the upper layer protocol is not dependent on a specific lower layer protocol for priority forwarding.

10

BRIEF DESCRIPTION OF THE DRAWINGS

15

The foregoing and other objects and advantages of the invention will be appreciated more fully from the following further description thereof with reference to the accompanying drawings wherein:

20

FIG. 1 shows an exemplary protocol stack that is used for TCP/IP networking;

FIG. 2 shows an exemplary protocol stack in accordance with a standard seven layer model;

FIG. 3 is a conceptual view of a portion of a protocol stack including an upper layer protocol that interfaces to a lower layer protocol via a high priority message queue, a low priority message queue, and a priority function in accordance with an embodiment of the present invention;

25

FIG. 4 is a logic flow diagram showing exemplary logic for installing an upper layer protocol above a lower layer protocol in a protocol stack in accordance with an embodiment of the present invention;

25

FIG. 5 is a logic flow diagram showing exemplary logic for forwarding a protocol message by the lower layer protocol to the upper layer protocol in accordance with an embodiment of the present invention; and

FIG. 6 is a logic flow diagram showing exemplary priority function logic in accordance with an embodiment of the present invention.

30

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

In an embodiment of the present invention, an upper layer protocol that requires priority forwarding includes a priority function that is invoked by a lower layer protocol in order to obtain the priority for a protocol message. When the upper layer protocol is installed in the protocol stack above the lower layer protocol, the upper layer protocol provides the priority function to the lower layer protocol, preferably by passing to the lower layer protocol a reference to the priority function (such as an address of the priority function). When the lower layer protocol receives a protocol message for the upper layer protocol, the lower layer protocol invokes the upper layer protocol's priority function and obtains therefrom a priority indicator indicating whether the protocol message is a high priority protocol message or a low priority protocol message. The lower layer protocol places the protocol message in the appropriate queue based upon the priority indicator obtained from the upper layer protocol's priority function. Specifically, if the priority indicator indicates that the protocol message is a high priority protocol message, then the lower layer protocol places the protocol message in the high priority message queue. If the priority indicator indicates that the protocol message is a low priority protocol message, then the lower layer protocol places the protocol message in the low priority message queue. In this way, the lower layer protocol is not required to interpret any protocol elements of the upper layer protocol in order to determine a priority for each protocol message, and the upper layer protocol is not dependent on a specific lower layer protocol for priority forwarding.

There are many different protocols and protocol layering schemes in use today.

FIG. 1 shows an exemplary protocol stack 100 that is used for TCP/IP networking. The TCP/IP protocol stack is considered to be a four (4) layer protocol stack, and includes a network interface layer 104, an internet layer 106, a transport layer 108, and an application layer 110. The network interface layer 104 is the interface between the TCP/IP protocol stack and the underlying hardware 102. The Internet Protocol (IP) typically operates at the internet layer 106. The Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) typically operate at the transport layer 108. Various routing

protocols, such as the Routing Information Protocol (RIP) and Open Shortest Path First (OSPF), typically operate at the application layer 110. The TCP/IP protocol suite includes many other protocols that operate at various layers of the stack.

FIG. 2 shows an exemplary protocol stack 200 that reflects the International Organization for Standardization (ISO) seven (7) layer model. The protocol stack 200 includes a physical hardware connection layer 202, a data link layer 204, a network layer 206, a transport layer 208, a session layer 210, a presentation layer 212, and an application layer 214. One common networking protocol, known as X.25, implements the lower three (3) layers of the ISO seven (7) layer model.

Other than a protocol operating at the lowest layer of a protocol stack, any protocol operating at any protocol layer may require priority forwarding. When installed as an upper layer protocol above a lower layer protocol, the upper layer protocol provides a priority function to the lower layer protocol, preferably by passing to the lower layer protocol a reference to the priority function (such as an address of the priority function). When the lower layer protocol receives a protocol message for the upper layer protocol, the lower layer protocol invokes the upper layer protocol's priority function and obtains therefrom a priority indicator indicating whether the protocol message is a high priority protocol message or a low priority protocol message. The lower layer protocol places the protocol message in the appropriate queue based upon the priority indicator obtained from the upper layer protocol's priority function.

FIG. 3 is a conceptual view of a portion of a protocol stack 300 including an upper layer protocol 302 and a lower layer protocol 310. The lower layer protocol 310 passes protocol messages to the upper layer protocol 302 over a high priority queue 304 and a low priority queue 306. The upper layer protocol 302 provides the priority function 308 to the lower layer protocol 310 when the upper layer protocol 302 is installed in the protocol stack above the lower layer protocol 310. The lower layer protocol 310 invokes the priority function 308 in order to determine a priority for each protocol message destined for the upper layer protocol 302.

FIG. 4 is a logic flow diagram showing exemplary logic 400 for installing an upper layer protocol above a lower layer protocol in a protocol stack. Beginning at step 402, the

logic establishes a message passing mechanism between the upper layer protocol and the lower layer protocol, in step 404, and provides the lower layer protocol with a reference to the upper layer protocol's priority function, in step 406. The logic 400 terminates in step 499.

FIG. 5 is a logic flow diagram showing exemplary logic 500 for forwarding a protocol message by the lower layer protocol to the upper layer protocol. Beginning at step 502, and upon receiving a protocol message for the upper layer protocol, in step 504, the logic determines whether the upper layer protocol is associated with a priority function, in step 506. If the upper layer protocol is not associated with a priority function (NO in step 508), then the logic forwards the protocol message to the upper layer protocol as a low (normal) priority message. However, assuming the upper layer protocol is associated with a priority function (YES in step 508), the logic invokes the priority function in order to obtain a priority indicator indicating a priority for the protocol message, in step 510. If the protocol indicator indicates that the protocol message is a high priority protocol message (YES in step 512), then the logic forwards the protocol message to the upper layer protocol as a high priority message, in step 514, for example, by placing the protocol message on a high priority message queue. If the protocol indicator indicates that the protocol message is a low priority protocol message (NO in step 512), then the logic forwards the protocol message to the upper layer protocol as a low priority message, in step 516, for example, by placing the protocol message on a low priority message queue. The logic 500 terminates in step 599.

FIG. 6 is a logic flow diagram showing exemplary priority function logic 600 in accordance with an embodiment of the present invention. Beginning at step 602, and upon being invoked by the lower layer protocol, the logic determines the protocol message priority based upon protocol-specific considerations, in step 604. If the logic determines that the protocol message is a high priority protocol message (YES in step 606), then the logic returns a high priority indicator to the lower layer protocol, in step 608. If the logic determines that the protocol message is a low priority protocol message (NO in step 606), then the logic returns a low priority indicator to the lower layer protocol, in step 610. The logic 600 terminates in step 699.

5

In the exemplary embodiments of the invention described above, the priority function is used to distinguish between two priority levels, specifically a high priority and a low priority. However, the present invention is in no way limited to two priority levels. The priority function may be used to distinguish between more than two priority levels, and more than two queues may be used for the lower layer protocol to forward the protocol messages to the upper layer protocol. The present invention is not limited by the number of priority levels distinguished by the priority function.

10 Also, in the exemplary embodiments of the invention described above, multiple queues are used to forward protocol messages from the lower layer protocol to the upper layer protocol. However, the present invention is in no way limited to the use of multiple queues for forwarding protocol messages. Other forwarding mechanisms may be used. For example, a single queue may be used for forwarding low priority protocol messages, while an interrupt- or event-based mechanism may be used to forward high priority protocol messages. The present invention is not limited by the mechanism or mechanisms used for forwarding protocol messages.

15

20 Also, in the exemplary embodiments of the invention described above, the upper layer protocol provides the lower layer protocol with a reference to the priority function, specifically by passing a pointer to the priority function. However, the present invention is in no way limited passing a pointer to the priority function. Other reference mechanisms may be used. For example, the upper layer protocol and the lower layer protocol may interface through a function table, in which case the upper layer protocol provides the lower layer protocol with a reference to the priority function by storing a pointer to the priority function in an appropriate location in the function table. The present invention is not limited to the mechanism by which the upper layer protocol provides the lower layer protocol with a reference to the priority function.

25

30 In an exemplary embodiment of the present invention, predominantly all of the priority forwarding logic is implemented as a set of computer program instructions that are stored in a computer readable medium and executed by an embedded microprocessor system within a network device. Various embodiments of the invention may be implemented in any conventional computer programming language. For example, an

embodiment may be implemented in a procedural programming language (*e.g.*, “C”) or an object oriented programming language (*e.g.*, “C++”). Alternative embodiments of the invention may be implemented using discrete components, integrated circuitry, programmable logic used in conjunction with a programmable logic device such as a Field Programmable Gate Array (FPGA) or microprocessor, or any other means including any combination thereof.

Alternative embodiments of the invention may be implemented as a computer program product for use with a computer system. Such implementation may include a series of computer instructions fixed either on a tangible medium, such as a computer readable media (*e.g.*, a diskette, CD-ROM, ROM, or fixed disk), or fixed in a computer data signal embodied in a carrier wave that is transmittable to a computer system via a modem or other interface device, such as a communications adapter connected to a network over a medium. The medium may be either a tangible medium (*e.g.*, optical or analog communications lines) or a medium implemented with wireless techniques (*e.g.*, microwave, infrared or other transmission techniques). The series of computer instructions embodies all or part of the functionality previously described herein with respect to the system. Those skilled in the art should appreciate that such computer instructions can be written in a number of programming languages for use with many computer architectures or operating systems. Furthermore, such instructions may be stored in any memory device, such as semiconductor, magnetic, optical or other memory devices, and may be transmitted using any communications technology, such as optical, infrared, microwave, or other transmission technologies. It is expected that such a computer program product may be distributed as a removable medium with accompanying printed or electronic documentation (*e.g.*, shrink wrapped software), preloaded with a computer system (*e.g.*, on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the network (*e.g.*, the Internet or World Wide Web).

The present invention may be embodied in other specific forms without departing from the essence or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive.

-10-

maintaining a plurality of queues, each queue corresponding to one of the plurality of priority levels; and

placing the message on the queue corresponding to the priority level.